Automatic Protection Switch Decision Engine

Related Application

This invention is related to an application filed on November 20, 1999, titled "A Method for Overcoming Faults in an ATM I/O Module and Lines Connected Thereto," which bears the Serial No. 09/444154.

Background of the Invention

This invention relates to ATMs and, more particularly to circumventing of faults in I/O modules of an ATM.

FIG. 1 presents a general block diagram of a conventional local ATM switch 100 with a connected I/O module 10, and conventional remote ATM switch 200 with a connected I/O module 20 (that may be of the same construction as that of module 10). Module 10 contains a line interface unit (LIU) 110 that is connected to fiber 210, and a line interface unit 120 that is connected to fiber 220. Fiber 210 is the "service" line, in the sense that it carries live data between I/O module 10 and I/O module 20. Fiber 220 is the "protection" line, in the sense that it is ready to assume the active communication function of line 210, should fiber 210 fail. Within module 10, LIU 110 is connected to framer 111, and framer 111 is connected to APS switch unit 130. Similarly, LIU 120 is connected to framer 121, and framer 121 is connected to APS switch unit 130. APS switch 130 is connected to ATM processing unit 140, and the output of ATM processing unit 140 forms the output of I/O module 10. This output is connected to ATM switch fabric 100. Elements 111, 121, 130 and 140 are connected to a control CPU 150.

Under normal operating circumstances, traffic from the service fiber (210) passes through LIU 110 and framer 111, and is applied to APS switch unit 130. The switch is set to pass this traffic to ATM processing unit 140 and thence, to ATM switch fabric 100. In the reverse direction, traffic flows from switch fabric 100 to ATM processing unit 140, and is bridged by APS switch unit 130 to both framers 111 and 121. That traffic is then transmitted out on both fibers 210 and 220. From the above it can be realized that protection fiber 220 carries signals that are identical to the signals carried in service line 210. The only difference is that APS switch 130 in I/O module 10 passes only the signal

of framer 111 to switch unit 140 and, similarly, I/O module 20 at the remote destination passes only the signal of framer 123 to switch unit 145.

When a failure occurs, for example, when fiber 210 is severed, CPU 150 gets an interrupt signal via line 151 from a detector in framer 111. In response thereto, the CPU takes recovery action. First, the CPU checks to determine whether the protection line (220) is in good operating order. Upon an affirmative determination, CPU 150 orders APS switch 130 to disconnect the path from line 210 toward ATM processing unit 140, and to connect the path from line 220 to ATM processing unit 140. CPU 150 also creates an APS signal and casts it onto line 220 through framer 121, toward I/O module 20. Framer 113 at I/O module 20 provides the received APS signal to CPU 160, and CPU 160 directs APS switch unit 135 to switch the signal arriving on fiber 220 to ATM processing unit 145.

Sub Q2 While an ATM constructed with I/O modules as shown in FIG. 1, and employed in the manner described above, is able to circumvent problems that originate in the fiber or the LIU, it nevertheless had a significant weakness. Use of the APS switch within the I/O module requires one to connect the service fiber and the protection fiber to the same I/O module. Consequently, a general failure in the I/O module brings down both the service path and the protection path. On first blush, it would appear that placing the APS switch off the I/O module, in a separate circuit board that is interfaced between the I/O module and the ATM switch, would solve the problem because it would allow the service fibers and the protection fibers to be connected to different I/O modules. Alas, current design ATMs do not have the physical room for inserting the circuit board that would serve as the switches for selecting I/O modules. Moreover, such a solution is quite expensive.

The aforementioned related application discloses an improved arrangement that operates in a novel manner by allowing the connection of the service fiber and the protection fiber to different I/O modules. The necessary switching for implementing this arrangement is achieved by closing and opening buffers in the I/O modules, as the need dictates, by cooperation between the CPUs on the I/O modules of the service and the protection lines and the ATM switch fabric. That is, the active line has its framer buffer open, while the standby line has its framer buffer closed. In the other direction, traffic is

multi-cast onto both the service and the protection lines by the ATM processing unit. In this manner, the protection fiber always contains information, ready to be switched from standby mode into active mode.

The above-described scenario of what happens when a fiber such as fiber 210 is severed is but one of the conditions that the decision logic within the CPUs of the I/O modules must account for before a decision is reached as to whether to close the buffer of the service I/O module and open the buffer of the protection I/O module, or vice versa. The more complete, actual, situation is that the decision logic is responsive to various different conditions that may exist in both the service and the protection I/O modules, as well as to a user-provided control signal from a controller that is coupled to switch fabric 100.

As for the conditions that may be present on the service and protection I/O modules, there is the SD (signal degraded) condition and the SF (signal failed) condition. As for the inputs applied by a user, they include a manual switching directive, a forced switching directive, a lockout directive, or a Release directive.

- A manual switching directive aims to assign the protection line to be the active line, and the service line to be the standby line when there are no fault conditions, or vice versa, and, for whatever reason, the operator wishes to make the desired assignment.
- A forced switching directive aims to switch a line to the active state even if that line is in a degraded (SD) condition.
- A lockout directive aims to assign the service line (only) to be the active line without any regard to what state the service line and the protection line are in.
- The release directive voids the other directives.

Hierarchically, from the highest priority concerns for the decision logic, to the lowest priority concerns for the decision logic, the order is: lockout, SF in the protection line, FS (forced switching), SF in the service line, SD in the protection line, SD in the service line, and lastly, manual switching.

Prior art arrangements account for all inputs and for all existing conditions through a software module that implements a state machine. Such a state machine is quite large. For example, in the Lucent Technologies GV2000 ATM switch, the

aforementioned state machine has about 40 states, and about 10,000 lines of C code. This can slow performance and certainly increases the cost of maintenance.

Summary of the Invention

An improved arrangement is attained with a very simple decision logic that, based on a comparison between two numbers that are created through the setting of bits in two registers, either directs the service line to be in the active state or in the standby state, and conversely, directs the protection line to be in the standby state or in the active state. The decision logic is embedded in a combination of a filter that either accepts or rejected applied stimuli, and a table that acts on accepted stimuli by the setting and resetting of bits in the two registers in accordance with a unique specification.

Brief Description of the Drawings

FIG. 1 illustrates a prior art ATM arrangement;

FIG. 2 presents an arrangement that comports with the principles of this invention;

FIG. 3 is a flow chart of one process for switching operations from the service fiber to the protection fiber; and

FIG. 4 is a table describing the action of a filter that accepts or rejects input stimuli;

FIG. 5 shows the location specifications for service and protection logic registers;

FIG. 6 shows a table that specified the locations that are set, or reset, in the registers shown in FIG. 5 in response to stimuli accepted according to the FIG. 4 table; and

FIG. 7 presents a flow chart of the method disclosed herein.

Detailed Description

FIG. 2 presents an illustrative ATM arrangement where the protection line and the service line are connected to different I/O modules. It shows an ATM switch 100 and associated I/O modules 30, and 40 and 50. Modules 30-50 differ from module 10 in that APS switch unit 130 is effectively not found in these modules. Illustratively, FIG. 2 has

one duplex span to the right of ATM switch 100 that includes a service line and a protection line, and two simplex spans that do not have protection lines. To the left of ATM switch 100 there are two simplex spans. The service line of the duplex span is connected from I/O module 30 to destination 1 via fiber 210. The protection line of the duplex span is connected from I/O module 40, also to destination 1, via fiber 230. Fiber 220 is connected to LIU 120 of I/O module 30 and it forms a simplex span to a destination 2. Similarly, fiber 240 is connected to LIU 124 of I/O module 40 and it forms a simplex span to a destination 3. Fibers 250 and 260 are connected to LIUs 116 and 126, respectively, of I/O module 50.

The following exposition considers only the operation of the duplex span. Before proceeding with this exposition, however, it may be noted that, as indicated above, each framer in the illustrative embodiment of FIG. 2 includes a detector to detect loss of signal or loss of framing. Each framer also includes a buffer that can be closed or opened, so as to block the buffer from outputting any signals, or to allow signals to flow out, respectively. The CPU of the I/O module provides the signal that controls the state of the buffer. For example, the state of the buffer in framer 111 is controlled by a signal that flows on bus 141.

During normal operating conditions, data flows through fiber 210 (the service line) and LIU 110 into framer 111. This data is transferred to ATM processing unit 140 and thence to ATM switch 100. The same data is also present in fiber 230 (the protection line) but this data is blocked by an appropriate control signal on bus 142. Thus, ATM switch 100 receives only one stream of data. Presuming that the data which does reach ATM switch 100 (from I/O module 30) is addressed to framer 117 in I/O module 50, ATM switch 100 makes the transfer, and the data flows to framer 117. Thence, the data flows to fiber 250 through LIU 116. In the reverse direction, two payload data streams are created from the data of framer 117 by use of a multicast integrated circuit that is already present in conventional ATM processing units (i.e., in unit 147). One of the streams is addressed to framer 111 in I/O module 30, and the other stream is addressed to framer 115 in I/O module 40. The two streams pass through ATM switch 100 and, thus, the information is delivered to framers 111 and 115 and flows out of fibers 210 and 230, respectively. The address information in ATM processing unit 147 is maintained in a

memory within the processing unit, which memory is populated by CPU 157. CPU 157 obtains this information from controller 200 that is connected to ATM switch 100 through ATM bus 201 (and in this manner is able to reach any of the I/O modules). Controller 200 maintains information for the entire switch regarding the I/O modules to which service fibers and associated protection fibers are connected.

When an SD or an SF condition is detected, for example, by framer 111, the framer sends a corresponding signal to CPU 150 on line 151 and, as in the prior art, CPU 150 takes corrective action. The corrective action process is depicted in FIG. 3.

As shown in FIG. 3, in block 301 CPU 150 creates a control cell that is addressed to CPU 156. Control then passes to block 302, where the created cell is forwarded to ATM switch 100 via the ATM bus. Switch 100 forwards the created cell to CPU 156, again via the ATM bus, in block 303. Finally, in block 304 CPU 156 makes decisions about what actions, if any, should be applied to the buffers of framers 111 and 115, and executes those decisions. If the decision is to close an open buffer in framer 111 and correspondingly to open a closed buffer in framer 115 then, one of two sequences of actions can be taken: either open the buffer of framer 115 first, or close the buffer of framer 111 first. Regardless of the sequence chosen (and the choice may be made based on the type of fault condition that exists) CPU 156 creates a control cell that is addressed to CPU 150, CPU 150 received the control cell and acts on the directive it contains, and CPU controls the buffer of framer 115 directly. The following discussion explains how those decisions of block 304 are arrived at.

In addition to receiving information from CPU 150, the decision logic in CPU 156 also has access to information from framer 115 and, therefore, knows whether there is an SD or and SF condition at the protection line. Further, CPU 156 receives user-requests signals from a user terminal (not shown) through controller 200 (as does CPU 150), and those user-requests specify either a lockout, a forced switch, a manual switch, or a Release directive.

In accordance with the principles disclosed herein and depicted in the flow chart of FIG. 7, the information from framer 111, framer 115, and controller 200 is applied to a decision filter 256 that is shown in FIG. 2 to be associated with CPU 156. Decision filter 256 records the most recent command from controller 200 (block 302), and develops an

"Accept" (e.g., logic 1) or "Reject" (logic 0) control signal, as a function of the remembered most-recent directive from the user, and the inputs from framer 111 and 115. Operationally, FIG. 7 shows the control process, which shows that commands from controller 200 are applied to block 311, where the most recent command is stored, and the controller 200 commands as well as the other stimuli are applied to block 312, where a Accept/Reject decision logic is effected under influence of the output of block 311. The Accept/Reject signal output of block 312 dictates whether an action is taken with respect to the stimuli to CPU 156. Specifically, when the output of decision filter 256 is not at logic level 0, action is taken with respect to registers 356 and 456 within CPU 156. Otherwise, no action is taken. Register 356 is the service line register SLR, and register 456 is the protection line register PLR. Each contains an 8 bit number, with the bit map defined as shown in FIG. 4.

The action taken is a setting of various bits in the SLR and the PLR registers, in accordance with the table shown in FIG. 5, based on the directives in the table of FIG. 6. Operationally, this is done in blocks 313 and 314 of FIG. 7. Once the appropriate bits in the SLR and PLR registers are set as specified above (effectively adding or deleting from the numbers stored in registers 356 and 456), a decision is made (in block 315) as to whether to open or close the buffers of framers 115 and 111, or vice versa, as follows:

```
If ((service line is active) and (SLR > PLR))
{
    switch service line to standby;
    switch protection line to active;
{
If ((protection line is active) and (PLR > SLR))
{
    switch protection line to standby;
    switch service line to active;
{
```

The above execution code is represented in FIG. 7 by code segments "execute 1" and "execute 2."

sub a4 The above discloses the principles of this invention for an arrangement like the one disclosed in the related application that was initially identified. It should be understood, however, that this invention is much broader, and is not limited to the disclosed embodiment. Illustratively, it can be applied to prior art arrangements for protecting service from fiber failures. Moreover, the control embodied in FIG. 7 can be installed the controller module, as well as in the IO modules, etc. Also, it should be understood that while the term "register" is used, and sometimes that designates a distinct hardware element, in the context of this invention the term includes any location in memory where data may be stored.